Congratulations! You passed!

Grade received 90% | Latest Submission Grade 90% | To pass 80% or higher

Go to next item

1. What does a neuron compute? 1 / 1 point

- A neuron computes the mean of all features before applying the output to an activation function
- A neuron computes a function g that scales the input x linearly (Wx + b)
- A neuron computes an activation function followed by a linear function $z = Wx + b$
- A neuron computes a linear function $z = Wx + b$ followed by an activation function

Expand

Correct
Correct, we generally say that the output of a neuron is a = g(Wx + b) where g is the activation function (sigmoid, tanh, ReLU, ...).

2. Which of these is the "Logistic Loss"? 1 / 1 point

- $\mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = \mid y^{(i)} - \hat{y}^{(i)} \mid^2$
- $\mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = max(0, y^{(i)} - \hat{y}^{(i)})$
- $\mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = \mid y^{(i)} - \hat{y}^{(i)} \mid$
- $\mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = -(y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}))$

Expand

Correct
Correct, this is the logistic loss you've seen in lecture!

3. Suppose img is a (32,32,3) array, representing a 32x32 image with 3 color channels red, green and blue. How do you reshape this into a column vector x? 1 / 1 point

- x = img.reshape((32*32,3))
- x = img.reshape((1,32*32,3))
- x = img.reshape((3,32*32))
- x = img.reshape((32*32*3,1))

Expand

Correct

4. Consider the following random arrays a and b, and c: 1 / 1 point

$a = np.random.randn(3,4)$ # $a.shape = (3,4)$

$b = np.random.randn(1,4)$ # $b.shape = (1,4)$

$c = a + b$

What will be the shape of c?

- c.shape = (1, 4)
- c.shape = (3, 1)
- c.shape = (3, 4)
- The computation cannot happen because it is not possible to broadcast more than one dimension.

Expand

Correct
Yes. Broadcasting is used, so row b is copied 3 times so it can be summed to each row of a.

5. Consider the two following random arrays a and b: 1 / 1 point

$a = np.random.randn(4, 3)$ # $a.shape = (4, 3)$

$b = np.random.randn(1, 3)$ # $b.shape = (1, 3)$

$c = a * b$

What will be the shape of c?

- The computation cannot happen because the sizes don't match.
- c.shape = (4, 3)
- The computation cannot happen because it is not possible to broadcast more than one dimension.
- c.shape = (1, 3)

Expand

Correct
Yes. Broadcasting is invoked, so row b is multiplied element-wise with each row of a to create c.

6. Suppose our input batch consists of 8 grayscale images, each of dimension 8x8. We reshape these images into feature column vectors $\mathbf{x}^j$. Remember that $X = \left[\mathbf{x}^{(1)}\mathbf{x}^{(2)} \cdots \mathbf{x}^{(8)}\right]$. What is the dimension of X? 1 / 1 point

- (8, 64)
- (8, 8, 8)
- (64, 8)
- (512, 1)

Expand

Correct
Yes. After converting the 8x8 gray scale images to a column vector we get a vector of size 64, thus X has dimension $(64, 8)$.

7. Consider the following array: 1 / 1 point

$a = np.array([[2, 1], [1, 3]])$

What is the result of $a * a$?

- $\begin{pmatrix} 4 & 1 \\ 1 & 9 \end{pmatrix}$
- The computation cannot happen because the sizes don't match. It's going to be an "Error"!
- $\begin{pmatrix} 5 & 5 \\ 5 & 10 \end{pmatrix}$
- $\begin{pmatrix} 4 & 2 \end{pmatrix}$

Expand

Correct
Yes, recall that * indicates element-wise multiplication.

8. Consider the following code snippet: 0 / 1 point

$a.shape = (4, 3)$

$b.shape = (4, 1)$

for i in range(3):

for j in range(4):

c[i][j] = a[j][i] + b[j]

How do you vectorize this?

- c = a + b

- c = a.T + b.T
- c = a + b.T
- c = a.T + b

Expand

Incorrect
No. The a[j][i] being assigned to a[i][j] indicates that we are using a.T.

9. Consider the following arrays: **1 / 1 point**

$a = np.array([[1, 1], [1, -1]])$

$b = np.array([[2], [3]])$

$c = a + b$

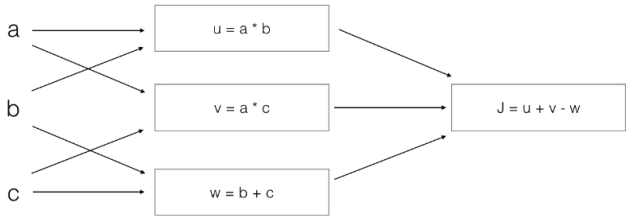
Which of the following arrays is stored in c?

- The computation cannot happen because the sizes don't match. It's going to be an "Error"!
- $\begin{matrix} 3 & 4 \\ 3 & 2 \end{matrix}$
- $\begin{pmatrix} 3 & 3 \\ 3 & 1 \\ 4 & 4 \end{pmatrix}$

Expand

Correct
Yes. The array b is a column vector. This is copied two times and added to the array a to construct the array c.

10. Consider the following computation graph. **1 / 1 point**



What is the output J?

- $J = (a - 1) * (b + c)$
- $J = (b - 1) * (c + a)$
- $J = (c - 1) * (b + a)$
- $J = a * b + b * c + a * c$

Expand

Correct
Yes. $J = u + v - w = a * b + a * c - (b + c) = a * (b + c) - (b + c) = (a - 1) * (b + c)$.